BAB II

TINJAUAN PUSTAKA

Teori merupakan dasar yang digunakan dalam penyusunan aplikasi yang dibangun. Pada bab ini akan dijelaskan mengenai tinjauan pustaka yang berkaitan erat dengan sistem informasi perpustakaan di SMU Pahlawan Toha.

2.1 Tinjauan Tempat Penelitian

Perpustakaan Sekolah adlaah perpustakaan yang diselenggarakan di sekolah-sekolah, yang bertugas mengumpulkan, menyimpan, memelihara, mengatur dan mendayagunkan bahan pustaka untuk kepentingan pengajar dan pendidikan disekolah tersebut. Saat ini di tiap sekolah pasti telah memiliki perpustakaa, dan sebagai slaah satunya adalah prpustkaan SMU Pahlawan, dibawah ini adalah profil dari SMU Pahlawan Toha :

Nama Perpustakaan	: SMU Pahlawan Toha
Alamat	: Jln. Peta Lingkar Selatan
Kode Pos	: 40233
Telepon	: (022) 606235

2.1.1 Sejarah SMU Pahlawan Toha

SMU Pahlawan Toha adalah salah satu SMU swasta di kota Bandung seperti pada umumnya sebuah SMU yang merupakan bangunan atau lembaga untuk belajar dan mengajar serta tempat menerima dan memberi pelajaran, sebuah sekolah dipimpin oleh seorang kepala sekolah, kepala sekolah dibantu oleh wakil kepala sekolah. Dalam sebuah sekolah ketersediaan sarana mempunyai peran penting dalam terlaksananya proses pendidikan, sarana tersebut meliputi ruang belajar, kantor, halaman/lapangan, dan perpustakaan.

Perpustakaan SMU Pahlawan Toha berdiri pada bulan Mei 1990, dan di bangun dari Perpustakaan seksi-seksi dan buku-buku hadiah yang semula hanya disimpan dibagian umum sekolah. Buku-buku tersebut dikumpulkan, diinventaris, diklasifikasi menurut bidang kajian berdasarkan jenis buku tersebut. Semua pekerjaan dikerjakan secara satu persatu merata dan oleh 2 orang karyawan.

Adapun dalam proses penyimpanan dan penyusunan koleksi perpustakaan atau klasifikasi menggunakan metode Dewey Decimal Classification (selanjutnya disingkat DDC). DDC adalah hasil karya Melvil Dewey (1851-1931), Secara umum DDC terdiri dari 3 (tiga) komponen utama yaitu:

· Bagan (schedules)

· Indeks relatif (relatives index)

· Tabel-tabel (tables)

1. Bagan (Schedules)

Bagan merupakan batang tubuh DDC. Didalam bagan ini semua ilmu disusun sedemikian rupa dan diberi kode angka yang disebut dengan notasi. Notasi dalam bentuk angka terdiri dari tiga angka. Apabila terdapat 4 (empat) angka atau lebih, maka antara angka ke-tiga dan ke-empat diberi tanda titik (.) seperti pada contoh 332.1 (Bank dan perbankan). Dengan prinsip desimal, DDC memberikan tiga ringkasan yang terdiri dari :

· 10 klas utama

· 100 divisi

· 1000 seksi dari bagan utama

Bila diinginkan masing-masing seksi dibagi pula secara desimal pada beberapa sub-seksi.

a. Klas Utama (10 ringkasan pertama), yaitu:

000 karya umum

100 Filsafat & Psikologi

200 Agama

300 Ilmu-ilmu sosial

400 Bahasa

500 Ilmu-ilmu murni dan matematika

600 Ilmu-ilmu terapan (Teknologi)

700 Kesenian, hiburan, olahraga

800 Kesusastraan

900 Geografi, Biografi dan Sejarah

b. Divisi (100 ringkasan kedua)

Setiap kelas utama dibagi secara desimal menjadi sub klas yang disebut "divisi" (100 ringkasan kedua)

Contoh: Misalnya diambil dari klas 300 (Ilmu-ilmu sosial):

300 Ilmu-ilmu sosial

310 Statistik umum

320 Ilmu politik

330 Ilmu ekonomi

340 Ilmu hukum

350 Administrasi negara (Pemerintahan), Ilmu militer

360 Layanan sosial, Asosiasi

370 Pendidikan

380 Perdagangan, Komunikasi dan Pengangkutan

390 Adat istiadat, Etiket dan Foklore

c. Seksi (1000 ringkasan ketiga)

Kemudian divisi ini dibagi menjadi 10 sub-divisi yang disebut 'seksi' (1000 ringkasan ke tiga)

Contoh : Misalnya diambil divisi klas 370 (Pendidikan)

370 Pendidikan

371 Sekolah

372 Pendidikan dasar

373 Pendidikan lanjutan

374 Pendidikan orang dewasa

375 Kurikulum

376 tidak digunakan lagi [unassigned]

377 tidak digunakan lagi [unassigned]

378 Pendidikan tinggi

379 Pendidikan dan Negara

Selanjutnya notasi tersebut dapat dibagi lagi secara desimal apabila dikehendaki.

Contoh : Misalnya diambil seksi klas 371 (Sekolah)

371 Sekolah

371.1 Pengajaran dan para pengajar

371.2 Administrasi dan manajemen sekolah

371.3 Metode mengajar dan belajar

371.4 Bimbingan dan penyuluhan

371.5 Disiplin sekolah

371.6 Sarana fisik (seperti; gedung, peralatan, laboratorium)

371.7 Kesehatan dan keamanan sekolah

371.8 Siswa

371.9 Pendidikan khusus atau Sekolah luar biasa

Dari contoh-contoh di atas dapat disimpulkan bahwa semakin khusus suatu subjek maka semakin panjang notasinya. (ringkasan klas utama dan divisi terlampir).

1. Jam Layanan

   Waktu pelayanan perpustakaan di SMU pahlawn Toha yaitu:

   - Senin – Kamis:
     07.30 – 16.00 WIB.
   - Jum'at:
     08.30 – 10.30 WIB.
     13.00 – 15.30 WIB
   - Sabtu:
     07.30 – 12.30 WIB

2. Sarana dan Prasarana

   Berikut sarana dan Prasarana yang dimiliki Perpustakaan SMU Pahlawan Toha antara lain:

Table2.1 : Sarana dan Prasarana Perpustakaan Pahlawan Toha

No.	Jenis Fasilitas
1	Ruangan
2	Luas ruangan
3	Meja baca
4	Rak buku 1 sisi
5	Rak buku 2 sisi
6	Kursi baca
7	Rak majalah
8	Meja komputer kary.
9	Kursi Karyawan

3. Personalia

Jumlah personil perpustakaan SMU Pahlawan Toha saat ini 2 orang yang mengerjakan seluruh kegiatan di perpustakaan.

4. Keanggotaan dan Fasilitas

Untuk menjadi anggota perpustakaan harus mempunyai kartu anggota perpustakaan. Syarat untuk mendapat kan katu anggota perpustakaan yaitu:

1) Siswa/i SMU Pahlawan Toha
    a. Menunjukan Kartu Pelajar yang berlaku
    b. Menyerahkan foto kopi 2x3 sebanyak 2 lembar
    c. Mengsi blangko untuk menjadi anggota.
    d. Peminjaman maksimal 3 eks selama 3hari.

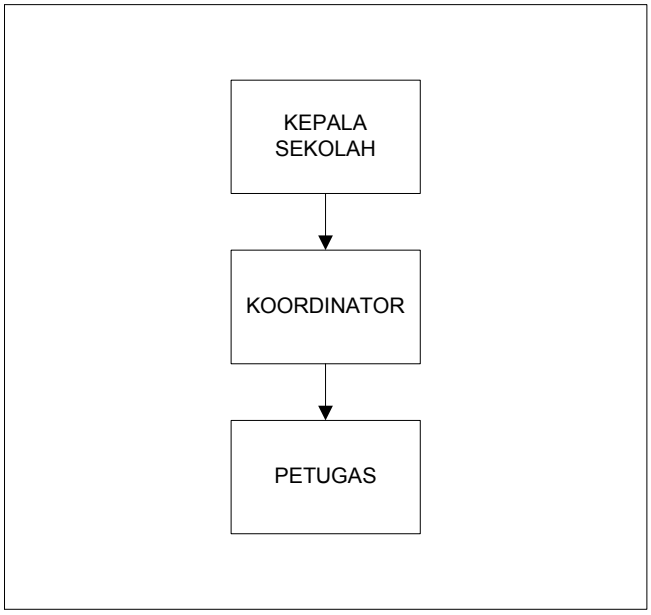
2) Sangsi
    a. Peminjaman buku yang terlambat mengembalikan buku di kenai sangsi berupa Denda perbuku perhari sebesar Rp.500,00
    b. Peminjaman yang menghilangkan buku wajib mengganti dengan buku serupa.
    c. Peminjaman buku harus sudah mengembalikan buku yang sebelumnya dipinjam.

2.1.2 Struktur Organisasi Pepustakaan SMU Pahlawan Toha

Struktur Organisasi dapat dikatakan sebagai pola hidup yang mapan diantara komponen-komponen atau bagian-bagian dari organisasi. Pada umumnya struktur organisasi berupa suatu gambaran grafis yang menunjukkan hubungan antara unit-unit serta wewenang yang ada.

Adapun organisasi itu berjalan dengan baik maka kerja sama diantara orang-orang dalam suatu organisasi sangatlah penting. Hal ini dikarenakan tenaga perorangan terbatas kemampuannya.

STRUKTUR ORGANISASI PERPUSTAKAAN SMU PAHLAWAN TOHA



Gambar 2.1 Struktur Organisasi Perpustakaan SMU Pahlawan Toha

2.1.3 Uraian Tugas.

Uraian tugas pokok dan rincian fungsi tiap-tiap bidang di SMU Pahlawan Toha adalah sebagai berikut :

1. Tugas Kepala Sekolah

a. Mengatur dan mengadakan urusan pendidikan.

b. Mengatur dan mengadakan urusan tata sekolah.

c. Mengatur dan mengadakan urusankeuangan sekolah.

d. Mengatur dan mengadakan urusan kepegawaian.

e. Mengatur dan mengadakan urusan perpustakaan.

f. Mengatur dan mengadakan urusan sarana dan perasarana

2. Koordinator

a. Mengawasi segal kegiatan yang ada diperpustakaan.

b. Mengatur urusan kepegawaian petugas.

c. Menjabarkan instruksi kepala sekolah.

3. Petugas

a. Mengatur dan mengadakan urusan pengadaan buku.

b. Mengatur dan mengadakan urusan pendaftaran anggota.

c. Mengelola proses peminjamandan pengemblaian buku.

d. Mengelola proses pelaporan data perpustakaan.

2.2 Konsep Dasar Sistem

Terdapat dua kelompok pendekatan yang dalam mendefinisikan sistem [2, hal1]. Pertama, pendekatan yang menekankan pada prosedur, mendefinisikan sistem sebagai berikut:

"Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu".
Kedua, pendekatan yang lebih menekankan pada elemen atau komponennya, mendefinisikan sistem sebagai berikut:

"Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai tujuan tertentu".

2.3 Konsep Dasar Informasi

Informasi (*informasi*) adalah data yang diolah menjadi bentuk yang lebih berarti bagi yang menerimanya [2, hal8]. Informasi juga dapat didefinisikan sebagai hasil pengolahan data sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan .

2.4 Konsep Dasar Sistem Informasi

Sistem informasi didefinisikan oleh Robert A.Leitch dari K R Davis sebagai berikut[2, hal 1]:

"Sistem informasi adalah suatu system didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi yang menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan".

John Buth dan Gani Drudnitski mengemukakan bahwa sistem inforamsi terdiri dari komponen-kompenen yang disebutnya dengan istilah blok bangunan(*building blok*), yaitu[2, hal 12]:

1. Blok masukan, input memiliki data yang masuk ke dalam sistem informasi.
2. Blok model, blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok keluaran, merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua pemakai sistem.
4. Blok teknologi, Terdiri dari tiga bagian utama, yaitu teknisi (*humanware atau brainware*), perangkat lunak(*software*) dan preangkat keras(*hardware*).
5. Blok basis data.
6. Blok kendali, beberapa pengendalian perlu darancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila trelanjur terjadi kesalahan-kesalahan dapat langsung tepat diatasi.

2.5 Metodelogi Pengembangan Sistem

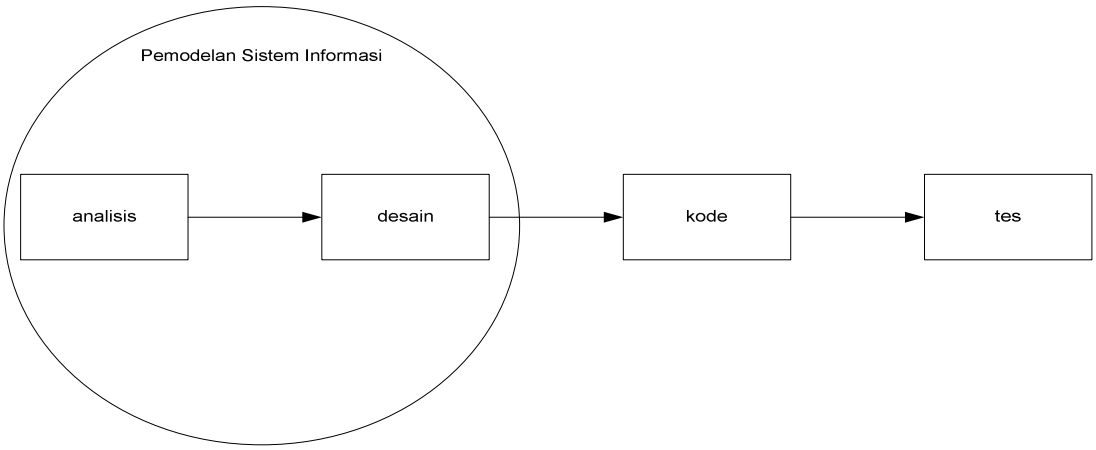
Sistem Development Lift Cycle (SDLC) adalah suatu metodologi yang digunakan untuk mengembangkan, memelihara, dan menggunakan sistem informasi (O'Brien,2001). Metodologi ini mencakup sejumlah fase atau tahapan.

Menurut Roger S.Presman, terdapat beragam model proses pengembangan perangkat lunak, diantaranya[4, hal36]:

1. *Linear Sequential* Model(Model Sequenstial Linear)

   Model ini pertama kali dikemukakan oleh Royce. Model ini sering disebut model klasik atau *waterfall*. Model ini menggunakan pendekatan pengembangan secara sekuen an sistematik untuk pengembanga perangkat lunak. Model ini merupakan model yang tertua. Model ini terdiri atas beberapa

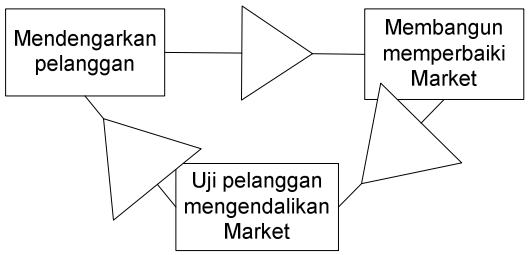
tahap yaitu: rekayasa dan prmodelan sistem/informasi, analisis kebutuhan perangkat lunak, desain, generasi kode, pengujian dan pemeliharaan.



Gambar 2.2 Model Sekuensial Linear

2. *Prototyping Model* (Model Prototipe)

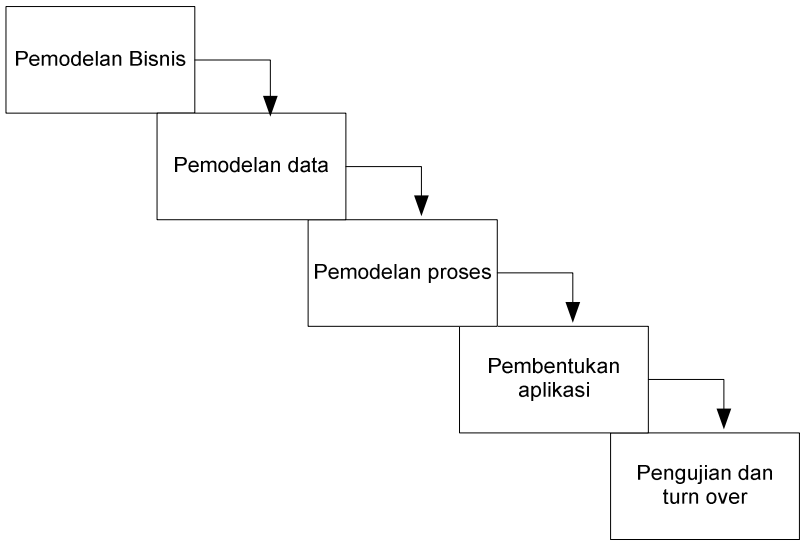
   Model Prototipe(*Prototype Paradigma*) dimulai dengan pengumpulan kebutuhan. Pengembang dan pelanggan bertemu dan mendefinisikan segala kebutuhan yang diketahui, dan area garis besar dimana definisi lebih jauh merupakan keharusan kemudian dilakukan perancangan kilat. Perangcangan kilat membawa kepada konstruksi sebuah prototipe yang kemudian di evaluasi oleh pelanggan/pemakai dan dipakai untuk menyaring kebutuhan pengembang perangkat lunak. Siklus dari model prototipe ada pada gambar 11.2.



Gambar 2.3 Model *Prototype*

3. *Rapid Application Development*(RAD) *Model*

RAD adalah sebuah model proses perkembangan perangkat lunak sekuensial linier yang menekankan siklus perkembangan yang sangat pendek. Model RAD merupakan adaptasi "berkecepatan tinggi" dari linier sequential model dimana pengembangan yang cepat dapat diperoleh dengan mengunakan pendekatan konstruksi berbasis kompoen.



Gambar 2.4 Model RAD

4. *Evalutionary Software Proses Model*

   Model evolusioner adalah model perulangan. Model ini dicirikan dengan pengembang mengembangkan versi-versi sistem yang lebih lengkap sedikit demi sedikit. Model telah mempertimbangkan untuk mengakomodasikan evolusi produk secara lengkap. Model ini terdi dari:

   a. *Incremental model*, model ini mengkombinasikan antara *linear sequential model* dengan filosofi iteratif pada *prototyping*. Pada masing-masing sekuen linear menghasilkan perangkat lunak yang semakin meningkat kompleksitasnya.
   b. *Spiral model*, model ini diusulkan oleh Boehm. Model ini menggabungkan antara sifat alami iterasi dari *prototyping* dengan aspek sistematik dan terkendali dari *linear sequential model*. Model ini memberi peluang untuk pengembangan cepat.
   c. Model rakitan komponen, model rakitan komponen menggabungkan beberapa karakter model spiral. Model ini bersifat evolusioner, sehingga membutuhkan pendekatan iteratif untuk menciptakan perangkat lunak. Tetapi model rakitan komponen merangkai aplikasi dari komponen perangkat lunak sebelum dipaketkan (kadang-kadang disebut ”kelas”).
   d. *Concurent development model*,

      Model perkembangan konkuren disebut juga rekayasa konkuren. Model proses yang konkuren dapat disajikan secara skematis sebagai sederetan aktivitas teknis mayor, tugas-tugas dan keadaannya yang lain.

5. Model Formal

   Model formal mencakup sekumpulan aktivitas yang membawa kepada spesifikasi matematis perangkat lunak komputer. Model ini memungkinkan perekayasa perangkat lunak untuk mengkhususkan, mengembangkan, dan memverifikasi sistem berbasis komputer dengan menggunakan notasi matematis yang tepat.

6. Teknik Generasi Keempat

   Bentuk “teknik generasi keempat” (4GT) mencakup serangkaian alat Bantu perangkat lunak yang kuat yang secara umum memiliki satu hal : masing-masing memungkinkan perekayasa perangkat lunak pada suatu tingkat yang tinggi. Alat bantu tersebut kemudian secara otomatis memunculkan kode sumber yang berdasarkan pada spesifikasi perekayasa.

Meskipun jumlah tahapan dalam SDLC dalam berbagai literature berbeda-beda, namun pada prinsipnya secara keseluruhan semua proses yang dilakukan sama saja. Tahapan-tahapan dalam SDLC adalah sebagai berikut:

1. Analisis Sistem, tahapan ini dimulai karena adanya permintaan terhadap system baru. Tujuan utama analisis sistem adalah untuk menentukan hal-hal detail tentang yang akan dikerjakan oleh sistem yang diusulkan (dan bukan bagaimana caranya). Analisis sistem mencakup studi kelayakan dan analisis kebutuhan.
2. Desain Sistem, tahapan ini dibagi kedalam dua subtahap, yakni perancangan konseptual dan perancangan fisik. Target akhir tahapan ini adalah menghasilkan rancangan yang memenuhi kebutuhan yang ditentukan selama

tahapan analisis sistem. Hasil akhirnya berupa spesifikasi rancangan yang sangat rinci sehingga mudah diwujudkan pada saat pemrograman.

3. Implementasi Sistem, pada tahap ini *programmer* harus mampu mengimplementasikan desain sistem kedalam bahasa pemrograman, untuk kemudian dilakukan pengujian.
4. Pengembangan dan pemeliharaan sistem, tahap ini dilakukan untuk mendeteksi kesalahan-kesalahan sistem yang tidak terdeteksi pada masa pengujian sistem.

2.6 Basis Data

2.6.1 Definisi Basis Data

Basis data (*database*) terdiri atas dua kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, konsep, keadaan, dan sebagainya. Yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya [1, hal2].

Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti [1, hal2]:

1. Himpunan kelompok data atau arsip yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redudancy*) yang tidak perlu untuk memenuhi berbagai kebutuhan.
3. Kumpulan *file* atau tabel yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

Basis data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data/arsip. Sedangkan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data/arsip. Perbedaannya hanya terletak pada media penyimpanan yang digunakan. Basis data bukan hanya sekedar penyimpanan data secara elektronis. Artinya, tidak semua penyimpanan data secara elektronis bisa disebut data. Yang sangat ditonjolkan basis data adalah pengaturan, pemilinan, pengelompokan maupun pengorganisasian data yang akan kita simpan sesuai fungsi dan jenisnya.

2.6.2 Operasi Dasar Basis Data

Operasi-operasi dasar yang dapat kita lakukan berkenaan dengan basis data dapat meliputi [1, hal4]:

1. Pembuatan basis data baru (*create database*)
2. Penghapusan basis data (*drop database*)
3. Pembuatan file/tabel baru ke suatu basis data (*create table*)
4. Penghapusan file/tabel dari suatu basis data (*drop table*)
5. Penambahab/pengisian data baru kesebuah file/tabel disebuah basis data (*insert*)

6. Pengambilan data dari sebuah file/tabel (*retrieve* atau *search*)
7. Pengubahan data dari sebuah file/tabel (*update*)
8. Penghapusan data dari sebuah file/tabel (*delete*)

2.6.3 Sistem Manajemen Basis Data

Manajemen Sistem Basis Data (*Database Managgement System* / DBMS) adalah perangkat lunak yang didesain untuk membantu dalam hal pemeliharaan dan utilitas kumpulan data dalam jumlah besar. DBMS dapat menjadi alternatif penggunaan secara khusus untuk aplikasi, semisal penyimpanan data dalam file dan menulis kode aplikasi yang spesifik untuk pengaturannya.

Penggunaan DBMS untuk suatu aplikasi tergantung pada kemampuan dan dukungan DBMS yang beroperasi secara efisien. Sehingga agar bisa menggunakan DBMS dengan baik, perlu diketahui cara kerja dari DBMS tersebut. Pendekatan yang dilakukan untuk menggunakan DBMS secara baik meliputi : implementasi DBMS dan arsitektur secara mendetail untuk dapat memahami desain dari suatu basis data[1].

2.6.4 Arsitektur Sistem Basis Data

Terdapat dua bentuk arsitektur sistem basis data, yaitu sistem terpusat dan sistem *client-server*.

Sistem basis data terpusat adalah sistem basis data yang dijalankan pada sistem komputer tungggal dan tidak berinteraksi dengan sistem pada komputer lain. Pengguna terkoneksi ke komputer pusat melalui terminal.

Sistem basis data *client-server* adalah sistem basis data yang memisahkan program program pengguna dengan program basis data disistem yang berbeda. Pengguna terkoneksi ke pusat data yang disebut server sistem melalui suatu program pengguna (*user interface*) yang terdapat pada *personal computer*. Sistem tempat program pengguna berada disebut *client system*.

2.7 Konsep Dasar Analisis Sistem

Analisis sistem (*System analysis*) dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, hambatan-hambatan yang terjadi dalam kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya [2, hal129]

Didalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analisis sistem sebagai berikut [2, hal 130]:

1. *Identify*, yaitu mendefinisikan masalah.
2. *Understand*, yaitu memahami kerja dari sistem yang ada.
3. *Analyze*, yaitu menganalisis sistem.
4. *Report*, yaitu membuat laporan hasil analisis.

Model analisis, merupakan representasi teknis yang pertama dari sistem. Saat ini ada dua yang mendominasi *landskap* pemodelan analisis, yaitu:

1. Analisis terstruktur. Dengan menggunakan notasi yang sesuai dengan prinsip analisis operasional, kita menciptakan model yang menggambarkan muatan dan aliran informasi (data dan kontrol), kita membagi sistem secara fungsional

dan secara *behavioral*, dan menggambarkan esensi dari apa yang harus dibangun. Analisis terstruktur bukan merupakan metode tunggal yang diaplikasikan secara konsisten oleh semua yang menggunakannya.

2. Analisis berorientasi objek, mendefinisikan semua kelas yang relevan terhadap masalah beserta operasi-operasi dan atribut-atribut yang diasosiasikan dengan kelas itu, keterhubungan dikelas-kelas dan perilaku yang dimilikinya. Sasssaran analisis berorientasi objek adalah untuk mngembangkan modal yang mendeskripsikan perangkat lunak yang memenuhi sekelompok kebutuhan yang didefinisikan pemesan. Analisis berorientasi objek menggunakan sejumlah pemodelan untuk memenuhi sasaran. Model analisis akan mengekspresikan informasi, perilaku dan fungsi didalam konteks model objek.

Alat-alat yang digunakan dalam analisis terstruktur adalah:

1. Diagram E-R (*Entity Relation Diagram*)
2. Diagram konteks (*Context Diagram*)
3. Diagram arus data (*Data Flow Diagram*)

2.7.1 Diagram E-R (*Entity Relationship diagram*)

Diagram E-R adalah diagram grafikal yang menggambarkan keseluruhan struktur *logic* dari sebuah basis data. Pada model ini semua data yang ada pada dunia nyata diterjemahkan atau ditransformasikan dengan memanfaatkan sejumlah perangkat konseptual menjadi sebuah diagram data.

Untuk melambangkan fungsi diatas maka digunakan symbol-simbol yang memiliki arti sebagai berikut:

a. Data Entitas

Entitas merupakan individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain. Data entitas adalah segala sesuatu baik yang nyata maupun abstrak yang datanya akan direkam.

b. Atribut

Atribut mendefinisikan karakteristik (*property*) dari entitas tersebut.

c. Relasi Data

Relasi menunjukkan adanya hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.

d. Kardinalitas

Kardinalitas adalah penentuan jumlah kejadian dari satu entitas yang dapat terhubung dengan sejumlah kejadian pada entitas lain, juga menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain.

Terdapat tiga kemungkinan kombinasi kardinalitas:

1. *Relationship one to one* (relasi satu ke satu)

   Yang artinya bahwa setiap entitas pada himpunan entitas yang satu berhubungan dengan paling banyak satu entitas pada himpunan entitas yang lainnya, begitu juga sebaliknya, entitas yang terhubung hanya memiliki satu hubungan dengan entitas yang menghubungi.

2. *Relationship one to many* (relasi satu banyak)

   Yang artinya bahwa setiap entitas pada himpunan A dapat berhubungan dengan lebih dari satu entitas pada himpunan entitas B, tetapi tidak berlaku

untuk kebalikannya, dimana entitas B hanya dapat berhubungan dengan paling banyak satu entitas pada himpunan A.

3. *Relationship many to many* (relasi banyak ke banyak)

   Yang artinya bahwa setiap entitas pada himpunan A dapat berhubungan dengan lebih dari satu entitas pada himpunan entitas B, begitu juga sebaliknya, dimana entitas B dapat berhubungan dengan lebih dari satu entitas pada himpunan A.

Objektif utama dari pembuatan Diagram E-R adalah untuk menunjukkan objek-objek (himpunan entitas) apa saja yang ingin dilibatkan dalam sebuah basis data dan bagaimana hubungan yang terjadi diantara objek-objek tersebut. Pada sebuah system yang ruang lingkupnya lebar dan kompleks, penggambaran atribut dalam sebuah Diagram E-R seringkali malah mengganggu objektif yang ingin dicapai. Kita dapat memisahkan pendeklarasian atribut-atribut ini dari Diagram E-R dan menyatakannya dalam sebuah kamus data berisi daftar atribut yang diapit kurung kurawal (‘{‘ dan ‘}’). Atribut yang berfungsi sebagai key juga dibedakan dengan yang bukan key dengan menggaris bawahi atribut tersebut [1, hal80].

2.7.2 Diagram Konteks

Diagram konteks adalah arus data yang berfungsi untuk menggambarkan keterkaitan aliran-aliran data antara system dengan bagian-bagian luar (kesatuan luar). Kesatuan luar ini merupakan sumber arus data atau tujuan data yang berhubungan dengan system informasi tersebut. Diagram konteks memberikan batasan yang jelas mengenai besaran-besaran entitas yang berada diluar system

yang sedang dibuat, artinya diagram ini menggambarkan secara jelas batasan-batasan dari sebuah sistem yang sedang dibuat.

Diagram konteks bisa disebut dengan "*Model sistem pokok (fundamental System Model) mewakili keseluruhan elemen software dengan masukan (input) keluaran (output) yang diidentifikasi dengan anak panah masuk dan keluar memperlihatkan sumber data*". (Roger S.Pressman, 1997)

2.7.3 Diagram Arus Data (*Data Flow Diagram*)

Data Flow Diagram adalah "*Teknik grafis yang menggambarkan aliran informasi dan perubahan yang dipergunakan sebagai perpindahan data dari masukan (input) keluaran (output)*". (Roger S.Pressman, 1997).

Beberapa simbol yang digunakan di DFD untuk maksud mewakili [2, hal 700]:

1. *External entity* (kesatuan luar)

   External entity (Kesatuan luar)merupakan kesatuan *(entity)* di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada dilingkungan luar sistem yang akan memberikan (*input*) atau menerima (*output*) dari sistem. Yang disimbolkan dengan kotak.

2. *Data flow* (arus data)

   Data flow (Arus data) mengalir diantara proses *(process)*, simpanan data *(data store)*, dan kesatuan luar *(external entity).* Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.

Yang disimbolkan dengan anak panah. Didalam menggambarkan arus data di DFD perlu diperhatikan beberapa konsep, yaitu:

a. Konsep paket dari data (*packet of data*), bila dua atau lebih data mengalir dari suatu sumber yang sama ketujuan yang sama, maka harus dianggap sebagai suatu arus data yang tunggal.
b. Konsep arus data menyebar (*diverging data flow*), arus data yang menyebar menunjukkan sejumlah tembusan dari arus data yang sama dari sumber yang sama ke tujuan yang berbeda.

3. *Process* (Proses)

Proses (*Proses*) adalah kegiatan atau kerja yang dilakukan oleh orang, mesin, atau komputer, untuk *Physical Data flow Diagram* (PDFD). Atau merupakan suatu proses yang hanya menunjukkan proses dari komputer, untuk *Logical Data Flow Diagram* (LDFD). Yang disimbolkan dengan lingkaran atau dengan simbol empat persegi panjang tegak dengan sudut-sudutnya yang tumpul.

4. *Data Storage* (Penyimpanan Data)

Simpan data (*data store*) merupakan tempat penyimpanan data. yang disimbolkan dengan sepasang garis *horizontal* yang tanpa tutup diujungnya.

2.7.4 Kamus Data

kamus data merupakan sebuah daftar yang terorganisasi dari elemen data yang berhubungan dengan sistem, dengan definisi yang teliti sehingga pemakai dan analisis sistem akan memiliki pemahaman yang umum mengenai *input*, *output*, komponen penyimpanan serta *kalkulasi intermediate*.

Pendefinisian data tersebut dilakukan dengan menggunakan notasi yang umum digunakan dalam menganalisa sistem yaitu dengan menggunakan sejumlah simbol. Kamus data biasanya dipelihara secara otomatis oleh sistem manajemen *database.*

Cara mendefinisikan kamus data yaitu :

1. Menggambarkan arti aliran data atau penyimpanan yang ditunjuk dalam DFD.
2. Menggabungkan komponen dari kumpulan data yang mengalir yaitu kumpulan komponen yang mungkin bisa dipecah lagi menjadi data elementer.
3. Menggambarkan data yang tersimpan.
4. Menentukan nilai dibagian elementer dari informasi yang relevan di DFD dan data *store*nya

2.8 Desain Sistem

Desain sistem didefinisikan sebagai berikut [1]*:*

"Penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi".

a. Tujuan Desain Sistem.

Desain sistem memunyai maksud dan tujuan sebagai berikut:

1. Untuk memenuhi kebutuhan kepada pemakai sistem.
2. Untuk memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada pemrogram komputer dan ahli-ahli teknik yang terlibat.

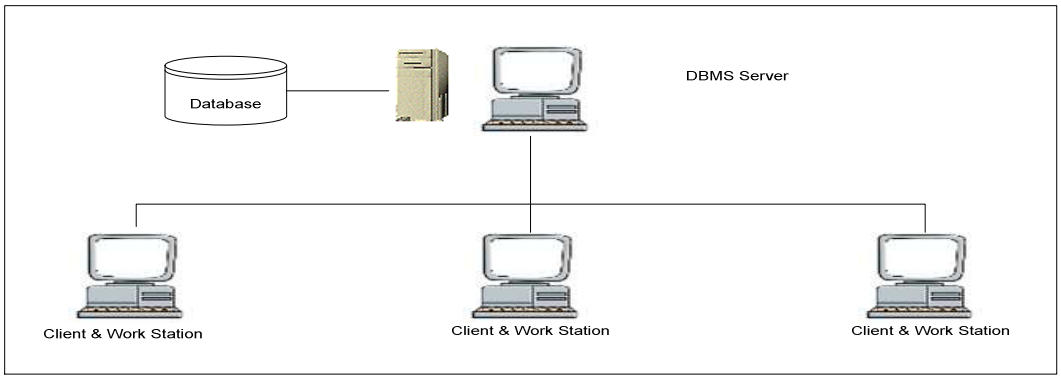
Sasaran-sasaran yang harus dicapai agar desain sistem mencapai tujuan:

1. Desain sistem harus berguna, mudah dipahami dan nantinya mudah digunakan.
2. Desain sistem harus dapat mendukung tujuan utama perusahaan.
3. Desain harus efektif dan efisien mendukung pengolahan transaksi yang, pelaporan manajemen dan mendukung keputusan yang akan dilakukan oleh manajemen, termasuk tugas-tugas lainnya yang tidak dapat dilakukan oleh komputer.
4. Desain sistem harus dapat mempersiapkan rancang bangun yang terinci untuk masing-masing komponen yang meliputi data dan informasi, simpanan data, metode-metode, prosedur-prosedur, orang-orang, perangkat keras, perangkat lunak dan pengendalian intern.

2.9 Sistem *Client Server*

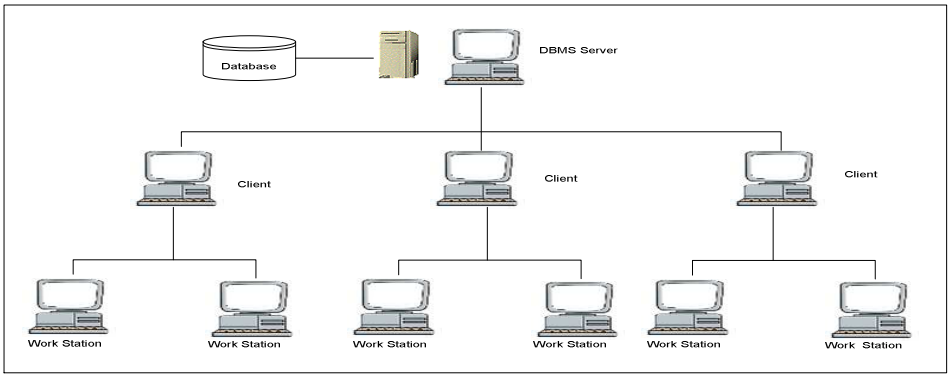
Sistem *Client Server* atau disebut juga sistem tersentralisasi diterapkan pada sebuah sistem jaringan. Sistem *Client Server* ini ditujukan untuk mengatasi kelemahan-kelemahan yang terdapat pada sistem sebelumnya. Sistem *Client Server* terdiri dari dua komponen utama yaitu *client* dan *server*. *Client* berisi aplikasi basis data dan *server* berisi DBMS dan basis data. Setiap aktivitas yang dikehendaki para pemakai akan lebih dulu ditangani oleh *client*. *Client* selanjutnya mengupayakan agar semua proses "sebisa mungkin" ditangani sendiri. Bila ada proses yang harus melibatkan data yang tersimpan pada basis data barulah *client* mengadakan hubungan dengan *server*.

Pada sistem *Client Server* untuk memenuhi kebutuhan, *client* akan mengirimkan *message* (perintah) *query* pengambilan data. Selanjutnya, *server* yang menerima *message* tersebut akan menjalankan *query* tersebut dan hasilnya akan dikirimkan kembali ke *client*. Dengan begitu transfer datanya jauh lebih efisien. Adapun bentuk dari Sistem *Client Server* yang sederhan adalah sebagai berikut :



Gambar 2.5. Sistem *Client Server* Sederhana

Disamping bentuk *client server* sederhana terdapat pula bentuk *client server* yang lebih komplek yang digamarkan sebagai berikut :



Gambar 2.6. Sistem *Client Server* Kompleks

Dari kedua gambar diatas, dapat dilihat adanya dua macam implementasi sistem *client server*. Bentuk yang sederhana dapat diterapkan pada sebuah jaringan komputer lokal (LAN) dimana fungsi *client* (untuk menangani sebagian besar proses pengolahan data seperti perhitungan, perulangan, pembandingan, dan lain-lain.) dan fungsi *work station* (untuk menangani interaksi dengan pemakai, menerima data masukan dan menayangkan hasil pengolahan) disatukan.

Adanya pemisahan fungsi *client* dan fungsi *server*, disamping meningkatkan kompleksitas tersendiri dalam pembangunan aplikasi secara keseluruhan, juga menimbulkan kelemahan lain, yaitu aktivitas pemasangan aplikasi yang tidak praktis. Bila terdapat perubahan / perbaikan aplikasi basis data maka harus mengulangi pekerjaan instalasi disemua mesin *client* yang digunakan. Karena itu pekerjaan ini sangat cocok diterapkan pada sistem jaringan yang lebar (WAN). Sedangkan pada varian sistem *client server* yang lebih kompleks, aplikasi basis data tidak ditempatkan disetiap *work station,* tetapi dipasang pada setiap *client* yang jumlahnya jauh lebih sedikit. Jadi setiap *client* dan sejumlah *work*

station membentuk sebuah LAN tersendiri. Karena *client-client* ini merupakan basis tempat aplikasi basis data disimpan dan turut menangani proses-proses dalam aplikasi, maka bagi *work station, client* ini dapat dipandang sebagai *server* aplikasi. Tidak bagaimana *work station* yang diaktifkan dan dinonaktifkan oleh para pemakai, *client-client* tersebut (sebagaimana juga DBMS *server)* harus selalu dalam keadaan aktif dan terkoneksi dalam sebuah jaringan yang lebih besar (WAN). Dengan begitu tahap instalasi aplikasi dapat dilakukan secara jarak jauh (*remote)* dari lokasi lain, sehingga kelemahan dari sisi instalasi dapat diatasi.

2.10 Tinjauan Perangkat Lunak

Dalam membangun sistem informasi diperlukan software pendukung yang menunjang dalam proses pembuatannya.

a. Borland Delphi

Dalam pembuatan suatu program aplikasi, beberapa rutin (prosedur dan fungsi) yang telah ada sering digunakan kembali. Cara konvensional untuk memanfaatkannya kembali adalah dengan mengkopikan rutin tersebut kedalam bagian program aplikasi yang dirancang, baik secara langsung maupun melalui direktif {$INCLUDE} ataupun melalui (DCU) yang disertakan dibagian uses. Tentunya hal ini menjadi kurang praktis kalau selalu harus mengkopikannya pada setiap program baru. Demikian pula jika ingin mengubah sifat-sifat objek tersebut dengan memberikan parameter. Walaupun bisa menambahkan suatu parameter untuk memenipulasinya, tetapi efeknya tidak dapat dilihat langsung pada saat

merancang programnya. Ini sangat berbeda dengan lingkungan pemrograman visual khususnya Delphi.

Dalam pemrograman visual seperti Delphi, salah satu kemudahannya (dan yang menarik) adalah tersedianya komponen visual (VCL, yang dikelompokan kedalam beberapa palet sesuai dengan klasifikasi atau fungsinya) lengkap dengan propertinya yang dapat dimanipulasi lebih lanjut. Cara menggunakannya pun sangat mudah, yaitu dengan menunjuk sebuah komponen, letakan pada *form* , dan atur propertinya.Komponen akan segera bereaksi terhadap perubahan nilai propertinya. Dengan demikian, dapat dikatakan bahwa komponen visual adalah objek pembentuk antar muka pengguna [4]. Dengan meletakan komponen pada form, akan otomatis membentuk antar-muka aplikasi.

Pada dasarnya apa yang tampak pada *form* akan sama dengan apa yang didapat saat program dijalankan. Ini adalah prinsip WYSIWYG (*What You See Is What You Get*), tentu tidak mutlak begitu. Sebagian efek manipulasi ini dapat dilihat langsung perubahannya pada tahap perancangan (melalui properti *run-time*) seperti *Visible* dan *Enable.* Yang paling menarik dari pemrograman visual ini adalah terbukanya kesempatan yang luas untuk merancang sendiri komponen tersebut. Bahkan, komponen visual yang telah ada dapat dikembangkan lebih lanjut untuk mendpatkan efek/ hasil tertentu sesuai dengan keinginan.

b. Microsoft SQL Server

Database adalah tempat penyimpanan data. *Database* tidak langsung menampilkan data ke user, tetapi user harus menjalankan aplikasi yang mengakses data dari *database* dan menampilkannya dalam bentuk yang mudah dimengerti. Untuk bekerja dengan database, kita harus memakai sebuah bahasa. Bahasa *database* yang paling banyak digunakan adalah SQL (*Strukture Query Language*). Microsoft SQL Server adalah sistem manajemen basis data yang memakai perintah-perintah *Transact*-SQL untuk mengirimkan perintah dari computer *client* ke server. *Transact*-SQL adalah bahasa SQL yang dikembangkan oleh Microsoft dengan menambah dialek- dialek tertentu. Microsoft SQL Server berisi *database*, mesin *database*, dan aplikasi yang diperlukan untuk mengolah data dan komponen- komponennya.

Tool - tool yang sering dipakai dan terdapat dalam Microsoaft SQL Server adalah:

1. SQL *Server Service Manager*, berguna untuk mengaktifkan atau menonaktifkan layanan (*service*) *server* SQL. *Client* hanya dapat melakukan transaksi database kalau server SQL telah diaktifkan .
2. *Query Analizer* , merupakan sebuah program berbasis grafis yang dapat digunakan untuk mendesain, mengetes dan menjalankan perintah - perintah SQL . Dalam *Query Analizer* terdapat dua bagian window yang penting yaitu bagian SQL yaitu tempat untuk menuliskan perintah SQL yang akan dijalankan dan window hasil eksekusi SQL yang menampilkan hasil eksekusi perintah SQL.

3. *Enterprise Manajer*, adalah program berbasis grafis yang berguna untuk adminisrasi *database*. Hal-hal yang dapat dilakukan dengan *Enterprise Manajer* adalah:
    a. Membuat *database*, tabel, *login* dan hak user terhadap server
    b. Pengisian data
    c. *Restruktur database*, tabel dan objek- objek lain.
4. *Book Online*, adalah sumber referensi utama yang memberikan informasi segala sesuatu mengenai SQL Server meliputi perintah SQL, fungsi-fungsi manipulsi data, dan lain-lain.
5. Impor dan Ekspor Data, berguna untuk mengkonversi data dari SQL Server ke *database* lain atau sebaliknya.